

# Huistaak - Lijsten

Nota:

1. indien 'register' in de declaratie van de lokale variabele of formele parameter staat, dan moet die variabele of parameter in een register worden bijgehouden (resp. doorgegeven); anders moet deze variabele op de stapel wordt bijgehouden (resp. via de stapel worden doorgegeven);
2. een functie geeft zijn resultaat terug via R0.

## Oefening 1

Vertaal.

```
int * iw;

main() {
    iw = alloc(1);
    printint (iw, *iw);
    *iw = 10;
    printint (iw, *iw);
    iw = (int *) 10;
    // verklaring van (int *):
    //   d.i. een type-conversie
    //   d.w.z. dat 10 als een adres moet beschouwd worden i.p.v. als een
    //   integer zonder deze conversie geeft de compiler een waarschuwing
    //   of een fout voor de vertaling naar DRAMA heeft dit geen enkel
    //   belang behalve wanneer je een adres zou willen gebruiken dat
    //   groter is dan 4999
    //   Bijv. iw = (int *) 8999; kan vertaald worden als: HIA.a Ri,8999
    //   iw = 8999; kan niet vertaald worden met directe operand
    printint (iw, *iw);
}
```

Voer dit programma uit met de simulator en verklaar de 6 getallen die afgedrukt worden.

## Oefening 2

Wat is de vertaling van het volgende C-programma? Het programma leest een aantal getallen in en bouwt een gesorteerde lijst op (van klein naar groot). De lijst heeft een kopelement.

```

struct elem {
    int info;
    struct elem * volg;
};

void inlassen (struct elem * l, int g) {
    register struct elem * p, *pp;
    struct elem * n;
    pp = l;
    p = pp->volg;
    while ((p != NULL) && (p->info < g)) {
        pp = p;
        p = pp->volg;
    }
    // Het nieuwe element toevoegen tussen pp en p
    // nota: in de volgende instructie geeft sizeof(...) de grootte aan van het
    // type. dit is GEEN functieoproep maar een opdracht voor de compiler!
    n = alloc(sizeof(struct elem));
    n->info = g;
    n->volg = p;
    pp->volg = n;
}

void druklijst (register struct elem * l) {
    l = l->volg; // sla kopelement over
    while (l != NULL) {
        printint(l->info);
        l = l->volg;
    }
}

main () {
    register struct elem * lst;
    int getal;
    // maak lege lijst (met kopelement)
    lst = alloc(2);
    lst->info = 0;
    lst->volg = NULL;
    // inlezen van positieve getallen
    getal = getint();
    while (getal > 0) {
        inlassen(lst, getal);
        getal = getint();
    }
    druklijst(lst);
}

```